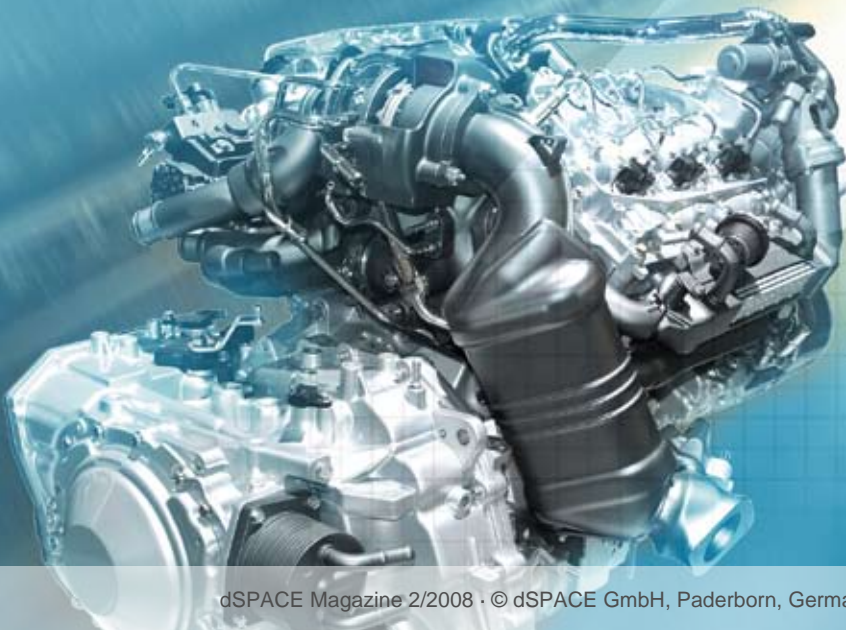


Engine Management AUTOSAR Way



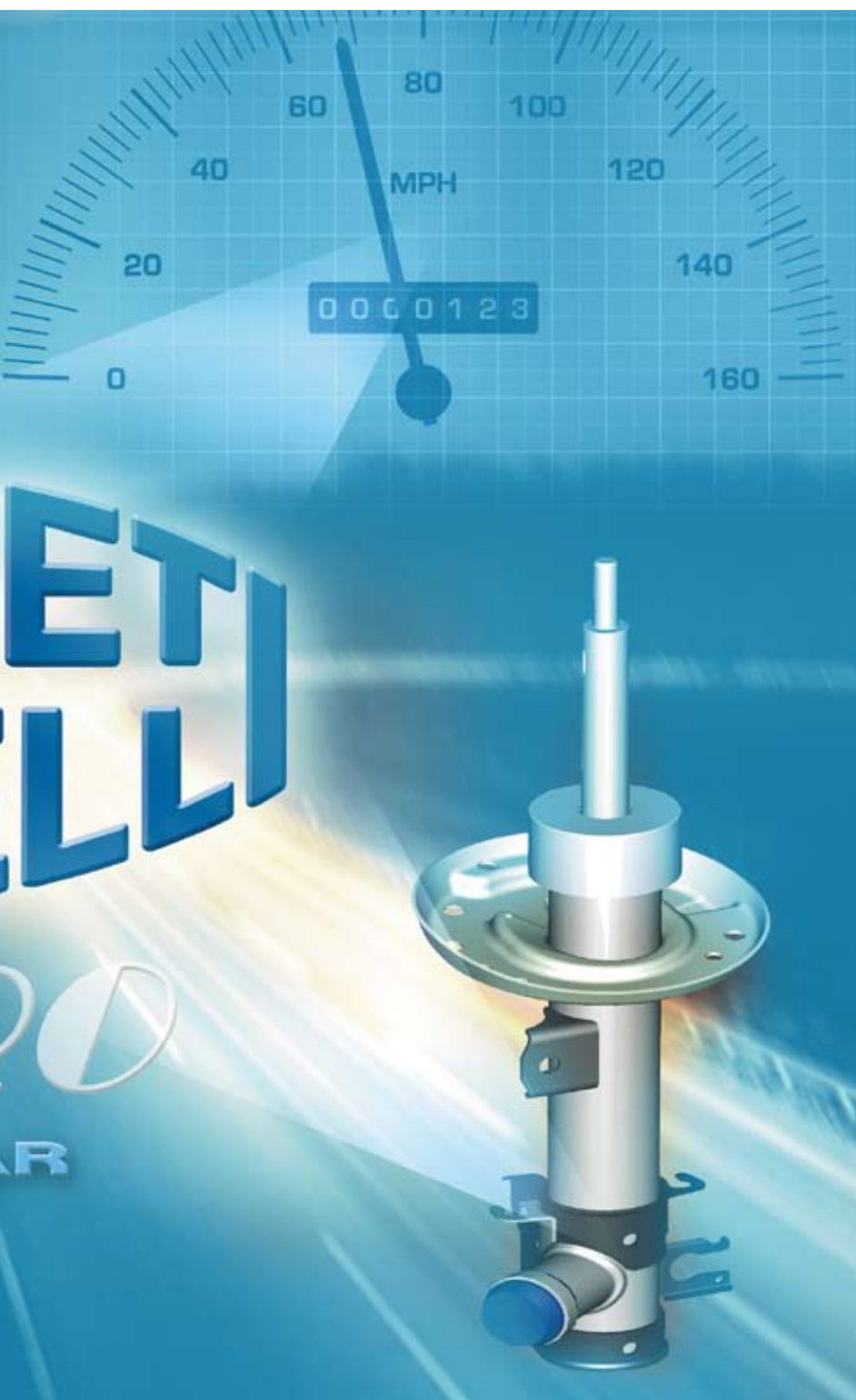
MAGN MARE

X-PP AUTOSAR



the

AUTOSAR 対応のエンジンマネジメントシステムの
開発 (Magneti Marelli 社)



AUTOSARは、自動車サプライヤと自動車メーカーの開発部門でクリティカルな話題となっています。この規格では、複雑なシステムを実装する際のオーバーヘッドが過大になるという話をよく耳にします。Magneti Marelli社は、AUTOSAR対応のエンジンECUプロジェクトを通じて、この内容が間違っていることを証明しました。

「Magnetis Marelli AUTOSAR Cross-Project X-PRO」は特別なプロジェクトで、当社ではAUTOSAR準拠のソフトウェアアーキテクチャの実現可能性を検証する専用のデモンストレータ（デモシステム）を開発しています。このソフトウェアアーキテクチャには、複雑なエンジン管理システムも含まれています。具体的な作業手順としては、既存のエンジンECUのソフトウェア全体をAUTOSARに移行してから、同じECUに再実装します。通常、エンジン制御ソフトウェアはサイズが非常に大きいので、各モジュールの再利用ができることはきわめて重要です。この点だけでも、AUTOSARによるソフトウェア開発は私たちに利益をもたらします。

dSPACEと共同で作業を行うため、私たちは、システム設計ソフトウェアであるSystemDeskの新しい自動化機能が主要な役割を果たすアプローチを選択しました。このデモンストレータプロジェクトは、実現可能性を検証すること、パフォーマンスおよびリソース消費の情報を収集することが目的であるため、まったく新しいアーキテクチャを設計することはせずに、大部分は既存のECUソフトウェア設計をそのまま利用しました。そのために、ソフトウェアアーキテクチャとスケジューリングの再構成に必要なすべての情報を既存のECUデータから抽出し、スクリプト（図1）を使用してSystemDeskに転送しました。この手順では、AUTOSAR準拠のアーキテ

クチャがほぼ自動的に生成されるので、SystemDesk側に必要な手作業による調整はごくわずかです。

データおよび情報の準備

AUTOSARが登場するまでは、すべての実装情報の記述に利用できる規格がなく、個別のソフトウェアモジュールを体系的に再利用することはできませんでした。したがって、私たちの最初の作業は、さまざまなデータソースからECUの構成、パラメータ設定、および実装のデータを収集し、それを後続の処理のために一つに集約することでした。たとえば、スクリプトを使用して、既存のTargetLink制御モデルからアーキテクチャの情報を抽出しました。また、ASAP2ファイルなどの記述

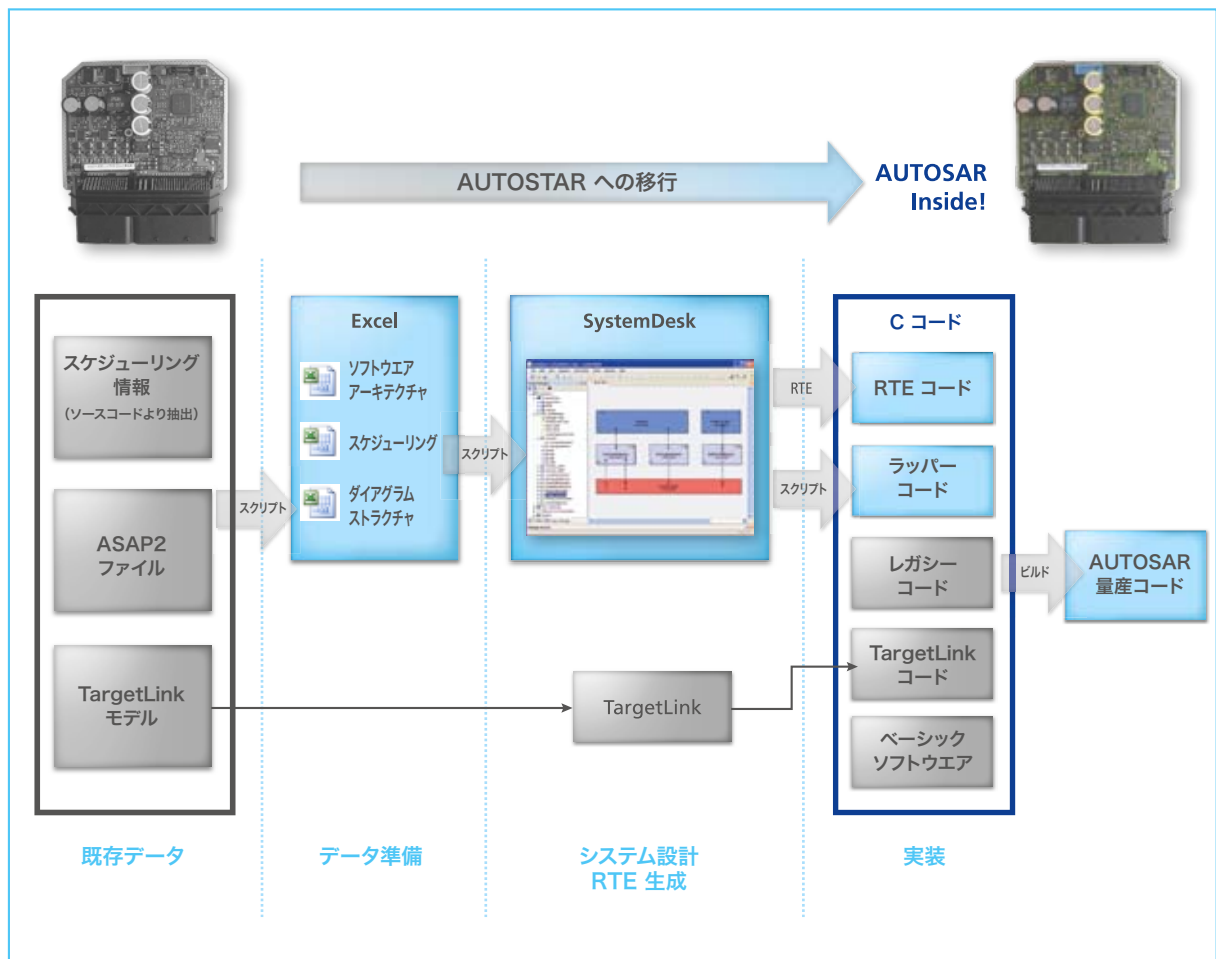


図1：AUTOSAR移行のための全体のワークフロー

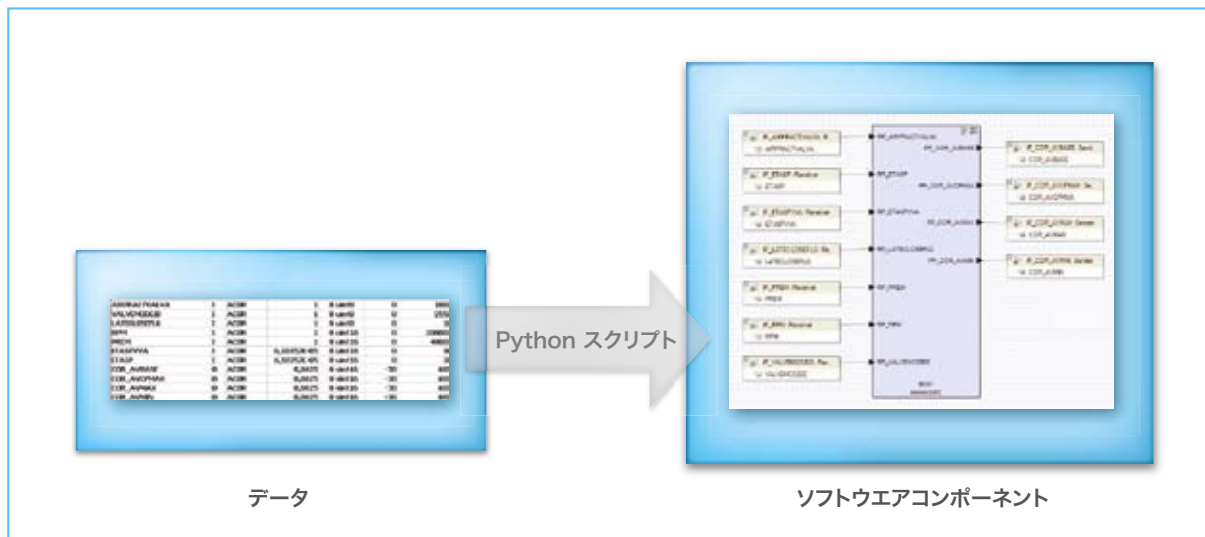


図 2 : Excel に収集されたソフトウェアコンポーネント (SWC) のすべての変数 (左) と、SystemDesk 内の対応する SWC (右)

ファイルを使用して、実装されている変数のデータ型、最小/最大値、およびスケールリングを取得しました。さらに、古い ECU のオペレーティングシステムの設定も再利用する必要があったため、スクリプトを使用して関数のスケジュール全体を古いソースファイルから抽出して、Excel に転送しました。

AUTOSAR のような規格により、将来的に既存のソフトウェアモジュールと実装データの再利用が非常に簡単になるのは明らかです。

ソフトウェアアーキテクチャの構築

次に Excel シートからのデータを実際のシステムに移行して、AUTOSAR 準拠のシステムアーキテクチャを作成する必要がありました。SystemDesk の自動化インターフェース経由で Python スクリプトを使用して、必要なソフトウェアコンポーネントのすべてを完全に自動的に作成しました。これには、センダ/レシーバインターフェースおよびデータ要素 (図 2) が含まれます。インポートする要素の数が 20,000 個を超えるような状況では、偶発的なエラーを完全に回避することはできません。しかし、SystemDesk では、入力データ内に存在

するすべての不整合を検出して表示できます。たとえば、データ型と指定されたスケールリング情報および最小/最大値が一致していない場合は、ソフトウェアアーキテクチャの作成中であってもこのエラーが表示されます。

変数をインポートするために、私たちは、

とめることができます。それぞれのダイアグラムは、ソフトウェアアーキテクチャのさまざまなビューを提供します。つまり、大規模なプロジェクトでも各担当者が何をしているか明確に把握できます。また複数のチームに関わる情報について議論することができます。

「SystemDesk で生成した RTE 経由での変数アクセスのパフォーマンスは、旧来の実装と変わりません」

Luigi Romagnoli, Magneti Marelli 社

ソフトウェアアーキテクチャ内のすべての接続を自動的に構築できる命名方式を使用しました。170 以上のソフトウェアコンポーネントおよび数千もの入力と出力信号がある場合、この命名方式を使用すると非常に多くの時間が節約できます。このようにデータ量が非常に多い状況では、一つのダイアグラム上にアーキテクチャ全体を表してもほとんど意味はありません。したがって、SystemDesk では、サブ機能をコンポジションダイアグラムでま

OS スケジュールのインポート

古いオペレーティングシステムスケジュールも、AUTOSAR ECU で使用する必要がありました。最初に Excel のシートを使用して Runnable をソフトウェアコンポーネントに割り当て、後はオペレーティングシステムタスク内での実行シーケンスを指定するだけでした。その後、スケジュールは完全に自動で SystemDesk に転送されました。

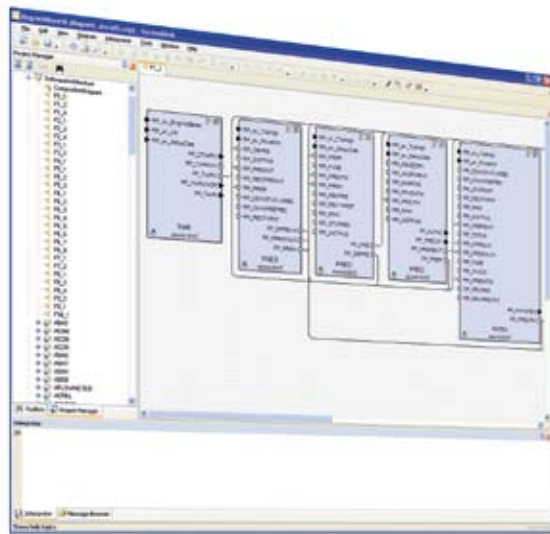


図3：個別のコンポジションダイアグラムに表示された全体モデルのサブセット

グラフィックによる複雑さの軽減

エンジン ECU は、車両に搭載されている ECU の中で最も複雑です。およそ 10,000 の変数、600 の Runnable、および 16 のタスクを定義して、最適に管理する必要があります。どのような方法にせよ複雑さを軽減できれば、エラーの防止および開発プロセスの短縮に役立ちます。既に説明したように、コンポジションダイアグラムではソフトウェアの個別の側面を表示することができますが、SystemDesk で複雑さを軽減する方法はこれだけではありません。他にも数多くの方法があります (図 3 も参照)。たとえば、3 回クリックするだけで、ソフトウェアコンポーネントの接続済みポートのすべてを非表示にすることや、ソフトウェアアーキテクチャ全体の未接続ポートのすべてを表示するダイアログを開くことができます。

SystemDesk のグラフィカルインターフェースの操作で特に優れているのは、複雑なシステムを簡素化された形式で表す機能です。これによってレビューを円滑に実行し、また新しいアイデアを画面上で素早く説明することができます。

RTE の生成

レガシーコードを AUTOSAR 規格に移行する場合、OS スケジュールを含むソフトウェアアーキテクチャ全体に対して RTE 生成を行うことが重要な中間点となります。ランタイム環境 (RTE) は、AUTOSAR ソフトウェアコンポーネント間の通信をソフトウェ

アで実現する簡素なミドルウェアです。状況によっては、この中間レイヤで、以前のソフトウェア実装よりも多くのリソースが必要になるかも知れません。したがって、SystemDesk によって生成される RTE コードのメモリと実行時間の要件を試算することが重要でした。試算に役立つ情報を得るために、その時点では、AUTOSAR に準拠したアプリケーションコードもベーシックソフトウェアも必要ありませんでした。

SystemDesk によって生成される RTE コードに関連するメモリ消費量を分析したところ、それは実質的に無視できるほどの量でした。変数へのアクセスはほぼ完全に define 文で実装されたので、旧来の実装と比較してパフォーマンスが犠牲になることはほとんどありませんでした。

アプリケーションコードの統合

RTE 生成で十分効率的なコードを実現できることは既実証されたので、現在私たちは、AUTOSAR アプリケーションレイヤの開発に取り組んでいます。このレイヤでは、手書きのレガシーコードを再利用するだけでなく、直接 TargetLink モデルから生成された過去の C コード全体も再利用します。コードを AUTOSAR 準拠にするために、私たちがプロジェクトのこの段階で採用しているアプローチでは、SystemDesk の自動化機能を使用して、レガシーコードと TargetLink コードの両方を AUTOSAR 準拠のラッパーに組込んでいます。この手法を選択したもう 1 つの理由は、レガシーコードと TargetLink コードの両方が同じインターフェースを持っているため、同じ種類のラッパーを使用できるからです。ただし、将来的には、TargetLink の AUTOSAR サポートを使用して、TargetLink モデルから AUTOSAR 準拠のコードとソフトウェアコンポーネント記述を直接生

「SystemDesk の自動化機能は、プロセスを短縮するだけでなく、データの整合性も確保します」

Alessandro Palma, Magnetis Marelli 社

「グラフィカルに複雑さを軽減するオプションが多数用意されているので、大規模なシステムモデルにも簡単に素早く対応できます」

Luigi Romagnoli, Magneti Marelli 社

成することを計画しています。このことにより、AUTOSAR 準拠のラッパーを不要にしてリソースの消費を改善し、また SystemDesk との相互作用におけるワークフローを簡素化することで、開発プロセス全体の効率が上がることが私たちは期待しています。

まとめと展望

私たちが実施した開発作業によって、既存のエンジン ECU の仕様および記述ファイルは、過度のオーバーヘッドもなく AUTOSAR に移行できることが明らかになりました。自動化可能なツールによる支援があれば、複雑なシステムや大量のデータにも確実に対応できます。初期パフォーマンスの測定では、開発ツールが最良の効率を発揮するように設計されていれば、AUTOSAR 規格を原因とする実行時間やメモリ消費の増加がないことが確認されました。AUTOSAR 準拠のアーキテク

チャおよびラッパーソフトウェアに関する作業は、既に完了しています。現在私たちは実装段階の作業を行っており、既存のアプリケーションコードとオペレーティングシステムの統合に取り組んでいます。今年の第 4 四半期中に、移行が完了した ECU を公開する予定です。■

Alessandro Palma
Luigi Romagnoli
Walter Nesci
Manager AUTOSAR Cross-Project X-PRO
walter.nesci@magnetimarelli.com
Magneti Marelli 社
イタリア

用語解説

ASAP2 – ASAP2 記述ファイルには、パラメータ、マップ、ルックアップテーブルのような特性など、ECU 内の関連データオブジェクトに関するすべての情報が格納されています。

Python – 非常にシンプルで実用性・汎用性に優れたスクリプト言語です。

Runnable – AUTOSAR SWC 内の実行可能な要素で、関数に相当します。

ランタイム環境 (RTE) – AUTOSAR アーキテクチャの中でアプリケーションコード内のソフトウェアコンポーネントとベーシックソフトウェアを接続する中間レイヤです。

スケジューリング – プロセスおよびタスク実行のタイミングを定義します。

センダ/レシーバインターフェース – AUTOSAR SWC 用のデータインターフェースです。

ソフトウェアコンポーネント (SWC) – 再利用可能なソフトウェアモジュールを作成するために使用する AUTOSAR の構成要素です。

タスク – システム内で動作するプロセスです。

現在のモデルデータ

- ソフトウェアコンポーネント：172
- データ要素：2650
- データアクセス：約 5000
- タスク：16
- Runnable：624

まとめ

- 既存の ECU を AUTOSAR へ円滑に移行
- 効率的な RTE コードを SystemDesk から直接生成
- エンジンマネージメントモデルの複雑さにも SystemDesk で対応可能